IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re patent application of:

Inventors:	Gadiel Seroussi et al.
Serial No.	09/916,785
Filed:	July 27, 2001
For:	Method and Apparatus for Random Bit-String Generation using Environment Sensors

Examiner: Jeffery L. Williams

Group Art Unit: 2137

Docket No. 10010554-1

Date: June 29, 2009

REPLY BRIEF UNDER 37 CFR 41.41(a)(1)

Mail Stop Board of Patent Appeals and interferences
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

In response to the Examiner's Answer dated April 28, 2009, applicant's reply as follows:

REAL PARTY IN INTEREST

The real party in interest is Hewlett-Packard Development Company, LP, a limited partnership established under the laws of the State of Texas and having a principal place of business at 20555 S.H. 249 Houston, TX 77070, U.S.A. (hereinafter "HPDC"). HPDC is a Texas limited partnership and is a wholly-owned affiliate of Hewlett-Packard Company, a Delaware Corporation, headquartered in Palo Alto, CA. The general or managing partner of HPDC is HPQ Holdings, LLC.

RELATED APPEALS AND INTERFERENCES

Applicant's representative has not identified, and does not know of, any other appeals of interferences which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

STATUS OF CLAIMS

Claims 10-13 are pending in the application. Claims 10-13 were finally rejected in the Office Action dated October 3, 2005. Applicant's appeal the final rejection of claims 10-13, which are copied in the attached CLAIMS APPENDIX.

STATUS OF AMENDMENTS

No Amendment After Final is enclosed with this brief. The last Amendment was filed July 27, 2006.

SUMMARY OF CLAIMED SUBJECT MATTER

Overview

Claims 10-14 are directed to a random generator shown in Figure 1 of the current application. As shown in Figure 1, digitally encoded input from one or more environmental sensors (11 in Figure 1) are input into one or more corresponding compressors (12 in Figure 1) that compress the received digitally encoded data from the sensors to generate compressed data streams. The compressed data streams are merged by a merge circuit (13 in Figure 1), which merges the compressed data streams received from the compressors, and monitors the resulting merged bit stream to ensure that sufficient bits are produced in the compressed streams to satisfy uncertainty requirements. A hash generator (15 in Figure 1) receives bits from the merged compressed data stream and generates an output block of bits that are output as next random number. The merge circuit (13 in Figure 1) controls a blocking switch (17 in Figure 1) to block output of a next random number until the merge circuit has received sufficient number of bits from the compressors that meet certain statistical requirements.

Independent Claim 10

Claim 10 is directed to a random number generation device such as that shown in Figure 1, described above. Claim 10 claims an environmental sensor (11 in Figure 1), a compressor (12 in Figure 1), a monitor (13 in Figure 1), a random number generator (15 in Figure 1), and a blocking switch (17 in Figure 1).

Dependent Claims 11-14

Dependent claim 11 adds additional sensors (11 in Figure 1) and compressors (12 in Figure 1). Independent claim 12 specifies that the random number generator (15 in Figure 1) applies a hash function to receive data to produce a next random number. Independent claim 13 further specifies that the environmental sensors (11 in Figure 1) may be selected from a number of different, specific types of environmental sensors, including temperature, sound, motion, light-intensity, and ambient-electromagnetic-radiation sensors.

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Whether claim 12 is indefinite under 35 U.S.C. §112, ¶2.

2. Whether claims 10-13 are unpatentable over Eastlake et al., "Randomness Recommendations for Security," R.F.C. 1750 ("Eastlake") in view of Saints et al., U.S. Patent No. 6,430,170 ("Saints").

ARGUMENT

Claims 10-13 are pending in the current application. In an Office Action dated October 12, 2005 ("Office Action"), the Examiner rejected claims 10-13 under 35 U.S.C. §112, ¶2 as being indefinite, and rejected claims 10-13 as being unpatentable over Eastlake et al., "Randomness Recommendations for Security," R.F.C. 1750 ("Eastlake") in view of Saints et al., U.S. Patent No. 6,430,170 ("Saints") under 35 U.S.C. §103(a). The current reply brief is filed in response to the Examiner's Answer, dated April 28, 2009. Appellants respectfully continue to traverse the 35 U.S.C. §112, ¶2 rejection of claim 12 and the 35 U.S.C. §103(a) rejections of claims 10-13, and continue to rely on the arguments and descriptions included in the appeal brief filed on April 24, 2007.

ISSUE 1

1. Whether claim 12 is indefinite under 35 U.S.C. §112, ¶2.

The Examiner does not mention the 35 U.S.C. §112, ¶2 rejection of claims 10-13 in the Examiner's Answer, and does not explicitly respond to Appellants' arguments with regard to these rejections. Appellants rely on the arguments provided in the originally filed appeal brief with regard to the 35 U.S.C. §112, ¶2 rejection of claims 10-13.

ISSUE 2

2. Whether claims 10-13 are unpatentable over Eastlake in view of Saints.

The remarks and arguments provided in this section are meant to address the points made by the Examiner in the lengthy Examiner's Answer dated April 28, 2009. Appellants continue to rely on the arguments and descriptions included in the appeal brief filed on April 24, 2007.

On page 4 of the Examiner's Answer, in the second paragraph following the centered title "Claim Rejections – 35 USC § 103," the Examiner cites section 9 of Eastlake, providing a summary of Eastlake's conclusion. The Examiner discusses low-quality random data, higher-quality random data, and processing higher-quality random data to produce usable random values. Appellants do not understand how this paragraph is related to the current claims, which do not include the language "low quality random data," "higher quality random data," and "useable random values." This paragraph is, in general, ambiguous and the phrases include relative and subjective terms, without definition.

In the final paragraph of page 4, the Examiner asserts that Eastlake discloses "an environmental sensor that generates digitally encoded sensor values," citing sections 5, 4.2, 5.3.1-2, 9, and the Abstract of Eastlake. Section 5 is a statement indicating that hardware vendors could provide a physical source of unpredictable numbers by adding a thermal noise or radioactive decay source and a free-running oscillator to computer systems. There is no mention, in section 5, of an environmental sensor. Section 4.2 discusses using mouse movement and key strokes to obtain random numbers, but suggests that these sources of random numbers are deficient and perhaps unusable. There is no mention of environmental sensors in section 4.4. Sections 5.3.1-2 mention uses of input of a sound digitizer with no

source plugged in or measurement of disk-seek time. In both cases, the data is related to internal components of a computer system, and not to environmental sensors that measure fluctuations in an environmental parameter, or, in other words, fluctuations in environmental conditions of the environment within which a device is located, as discussed in the final paragraph of page 3 of the current application. The thermal noise mentioned in section 5.3.1 is generated by internal resistance heating within a camera or microphone. There is no mention of environmental sensors in sections 5.3.1-2. Section 9 and the Abstract contain nothing related to environmental sensors. Neither section uses the phrases "random bit stream," "digitally encoded values," or "bit string." The Examiner would appear to read "environmental sensor" onto any type of measuring device, including devices that measure mouse clicks. This interpretation essentially ignores the claim term "environmental." The Examiner is not free to disregard claim terms.

In the first paragraph of page 5, the Examiner asserts that Eastlake teaches "a compressor that receives the digitally encoded sensor values generated by the environmental compressor and compresses the received digitally encoded sensor values to generate a compressed data stream," citing sections 5.2 – 5.31 and 5.3.4 of Eastlake. There is no section 5.3.4 in Eastlake. Eastlake discusses use of reversible compression to de-skew a bit stream in section 5.2.4. Section 5.2.4 is the only section of the cited sections which mentions compression. This section does not indicate the source of the bit stream that is de-skewed, exactly what the output of the de-skewing operation is, and how the output is used. In section 5.2.1, a bit string is disclosed as being de-skewed by computing the parity of the bit string, and mapping the bit string to either 0 or 1. Clearly, de-skewing does not necessarily produce multiple bits, and certainly does not necessarily produce a bit stream. Section 5.2.4 suggests producing a compressed sequence. A compressed sequence is not a compressed data stream. A sequence is, like a string of bits, a static collection that does not change over time. By contrast, a stream is a time sequence. A bit stream produces bits at time intervals. The value of a bit string can be generated by collecting bits from a bit stream, over time, but a bit stream is not a bit string. Section 5.2.4 states that use of reversible compression to de-skew a bit stream is at best a "rough technique," and that skipping of initial bits is required. There is no suggestion that the de-skewing operation produces a compressed bit stream, but, instead, Eastlake appears to suggest that a bit stream is input to the de-skewing process, and that some number of random bits are output, the random bits selected from a "compressed sequence" output by the compression technique. There is no mention of environmental sensors in

Eastlake.

Not finding any teaching, mention, or suggestion of the elements of claim 1 "a monitor that receives the compressed data stream and monitors the compressed data stream to determine whether or not sufficient data has been received in the compressed data stream to generate a next random number" and "a blocking switch controlled by the monitor to block output of a next random number by the random number generator when sufficient data to generate the next random number has not been received in the compressed data stream to generate a next random number," the Examiner turns to Saints, on page 6 of the Examiner's Answer. The Examiner states, towards the bottom of page 6, that Saint's random number generator unit "receives random bits resulting from the measurement of thermal noise by environmental sensors," citing lines 26-31 of column 5, lines 1-7 of column 7, and line 66 of column 5 to line 14 of column 6 of Saints. The Examiner is incorrect in this assertion. The input to the random number generation unit shown in Figure 4 is a collection of energy samples produced by Saints' searcher (200 in Figure 2). Saints states that the searcher outputs energy samples computed from an incoming radio signal. Saints does not teach, mention, or suggest that the energy samples are, in any way, random. No one familiar with radio signals would expect energy samples collected from a radio signal to be random.

The Examiner asserts that the "gathered random data is processed to ensure that the random data bits are of an acceptable quality," citing lines 1-7 of column 9. This assertion is incorrect. Lines 1-7 of column 9 discuss the fact that energy samples, which are clearly not random, are tested to reject easily guessed numbers, such as numbers with all 1's or 0's.

The claim language to which the above-mentioned incorrect assertions appear to be related follows: "a monitor that receives the compressed data stream and monitors the compressed data stream to determine whether or not sufficient data has been received in the compressed data stream to generate a next random number." Saints does not compress the energy samples, and therefore Saints cannot possibly disclose a monitor that monitors a compressed data stream. Saints does not monitor a compressed data stream "to determine whether or not sufficient data has been received in the compressed data stream to generate a next random number," but instead examines energy-sample input numbers in order to reject numbers with easily guessed values. The cited portions of Saints having nothing whatsoever to do with "a monitor that receives the compressed data stream and monitors the compressed data stream to determine whether or not sufficient data has been received in the compressed

data stream to generate a next random number."

The Examiner's comments on page 7 make no sense, and represent a rather serious misreading of Saints and an attempt to read clear claim language onto a completely unrelated disclosure. In column 9 of Saints, Saints describes operation of the internal components of Saints' random number generation unit. The unit is quite straightforward. At power up, and after a pool-refresh call, energy samples are loaded directly into the random pool buffer (406 in Figure 4; column 9, lines 8-9). Then, the first hash unit receives additional energy samples, and mixes the additional energy samples with values already residing in the pool, overwriting the pool with the mixed values. The counter, mentioned first on line 30 of column 9, is used during the process of initially mixing the pool with additional energy samples, to ensure that the pool is completely filled with mixed samples. It is the mixing operation that generates pseudo-random values. The energy samples are not random, and not claimed to be random by Saints. The process described from line 8 of column 9 to line 56 of column 9 is related to pool initialization. Once the pool is initialized, then random-number generation proceeds without further interruption, as discussed in the first paragraph of column 10. During random-number generation, bytes are removed from the pool for random-number generation, and new energy samples are mixed into the pool, using the pool as a ring buffer.

There is no blocking switch taught, mentioned, or suggested by Saints. The final element of claim 1 reads: "a blocking switch controlled by the monitor to block output of a next random number by the random number generator when sufficient data to generate the next random number has not been received in the compressed data stream to generate a next random number." Note that, in claim 1, the monitor that controls the blocking switch is "a monitor that receives the compressed data stream and monitors the compressed data stream to determine whether or not sufficient data has been received in the compressed data stream to generate a next random number." There is no compressed data stream in Saints and nothing that determines whether or not sufficient data has been received in the compressed data stream to generate a next random number. Saints' pool may contain far in excess of the amount of data needed to generate a single random number. Saints mentions nothing about sufficiency of data to generate a random number. Instead, Saints initializes a pool, on power-up, before launching on-demand generation of random numbers. The statement used by the Examiner to justify reading monitors and blocking switches into Saints random number generation unit, even though Saints fails to teach, mention, or even remotely suggest the

currently claimed monitor and blocking switch, is: "In a preferred embodiment, as soon as the pool is filled, a random number will be extracted from the pool" (column 9, lines 57-58). The statement simply discusses an ordering of operations. Saints does not state, or even remotely suggest, that the pool needs to be completely filled in order to generate a random number, but merely that, in Saints implementation, a random number is extracted from the pool once the pool is filled. Indeed, it is apparent that 20 bytes of data are extracted from the pool and input to the second SHA-1 unit (408 in Figure 4) for generation of a next random number, and that the pool is used as a ring buffer for this purpose (see column 10, lines 40-54). Saints does not make even slight mention of any kind of monitoring for determining whether sufficient data has been received to generate a next random number. Since the pool is used as a ring buffer, the pool contains far more data than needed to generate a single random number. Saints would not need to wait for the pool to fill before generating a next random number. Random-number generation could start well before filling the pool. There cannot be a monitor that operates as the current claimed monitor in Saints, because random number generation would begin before filling of the pool were the currently claimed monitor to control a blocking switch that, in turn, controls random number generation.

A combination of Eastlake and Saints, neither of which teaches, mentions, or suggests either "a monitor that receives the compressed data stream and monitors the compressed data stream to determine whether or not sufficient data has been received in the compressed data stream to generate a next random number" or "a blocking switch controlled by the monitor to block output of a next random number by the random number generator when sufficient data to generate the next random number has not been received in the compressed data stream to generate a next random number" cannot possibly make obvious the invention to which claim 1 is directed.

The Examiner, on page 11, requests clarification of the term "stream." A stream is a time sequence. A stream of water, for example, is a sequence, in time, of volumes of water flowing past a point or through a fixed cross-sectional surface within the stream. A stream of bits is a sequence of bits produced over a time interval. The word "stream" implies flow, which, in turn, involves a sequence in time. A string of bits is not a stream of bits. A string of bits can be collected, in memory, from the output of a stream of bits. Environmental sensors produce streams of data, measuring environmental parameters over time and produce output continuously over time.

The Examiner further discusses the term "compression," on page 2. The

Examiner's proposed definitions do not properly define this term. Data compression is discussed beginning on line 10 of page 4 of the current application. Data compression is not simply decreasing the size of a data set. Otherwise, for example, truncating a file to size 0 could be called "data compression," but the information in the file would be lost by truncating the file. In computer science and information theory, data compression involves removal of redundant data from a data set to decrease the size of the data set while preserving the original information content. There are lossless and lossy data compression techniques. In the first case, the original data can be exactly regenerated from the compressed data. In the latter case, the original data can only be approximately regenerated. Data compression is a very well-known operation to computer scientists and information theorists.

Next, the Examiner addresses previous arguments made by Appellants, beginning on page 12. With regard to section (i), claim 1 includes the elements: "a compressor that receives the digitally encoded sensor values generated by the environmental compressor and compresses the received digitally encoded sensor values to generate a compressed data stream" and "a monitor that receives the compressed data stream and monitors the compressed data stream to determine whether or not sufficient data has been received in the compressed data stream to generate a next random number." A stream, as discussed above, represents a flow of data, in time. A compressed data stream is a flow of compressed data. The currently claimed invention repeatedly taps into a stream of compressed data to extract a next quantity of data that is used to generate a next random number. This is a continuous process. Eastlake does not teach, mention, or suggest a stream-like flow of compressed data into a random number generator. Instead, Eastlake teaches use of a reversible compressor to de-skew a skewed bit stream to produce a string of bits, and then uses the string of bits as a seed value. Once the seed value is obtained, comprising around 100 random bits, random numbers are generated by strong computational techniques, as clearly stated in section 9 of Eastlake. Eastlake does not generate each random number from a compressed data stream, but, instead, uses a compressed data stream to generate a seed value for a random number generator. The random number generator is repeatedly called to generate a sequence of random numbers, without further need for compressed data streams. Eastlake's method is very different from that to which claim 1 is directed. Please note that Eastlake, in section 9 repeatedly cited by the Examiner, clearly states: "Once a sufficient quantity of high quality seed key material (a few hundred bits) is available, strong computational techniques are available to produce cryptographically strong ***sequences*** of

unpredictable quantities from this seed material" (emphasis added). This statement clearly summarizes Eastlake's approach. There is no suggestion in Eastlake for continuously generating random numbers from data extracted from a compressed data stream. Again, a compressed data stream is a flow of compressed data in time. The current claims are clearly directed to a continuous process, which is one reason that embodiments of the current invention employ environmental sensors, which each produces a continuous stream of data.

With regard to section (ii), the Examiner appears to have failed to appreciate the teaching in Eastlake. De-skewing a bit stream is an operation used to provide a uniform distribution, as explained in section 5.2. Eastlake then discusses various ways to de-skew skewed strings or streams of bits. Use of a reversible compressor is one way to do this. Eastlake discusses generation of a compressed sequence, from which a number of random bits are selected. That is all section 5.2.4 discloses. However, section 9 clearly indicates that those random bits are used, by Eastlake, merely as a seed for a random-number generator. The Examiner's rejections are largely based on assumptions and inferences, rather than teachings in Eastlake. Eastlake teaches nothing about extracting data from a compressed data stream to generate each next random number.

With regard to section (iii), the Examiner is attempting to substitute the Examiner's incorrect understanding of Eastlake for Eastlake's disclosure. Eastlake does not teach, in section 5.2.4 anything more than obtaining a string of random bits as a result of de-skewing a bit stream. Eastlake does not teach, mention, or suggest generating a compressed stream. The Examiner may consult the final sentences of section 5.2.4 to appreciate this fact. Eastlake uses compression only to generate a random bit *string*, and teaches nothing about generation of bit streams. Furthermore, Eastlake clearly teaches a need for random-number-generator seeds, but discloses no use for a compressed data stream. The Examiner's statements appear to reflect a desire to read the current claims onto unrelated subject matter, rather than finding a teaching or suggestion of the claimed subject matter.

With regard to section (v), Appellants have never asserted that the current claims include, or are directed to, "the large number of uncorrelated sources are compressed data streams." Appellants do not understand the Examiner's statements in this section.

With regard to section (vi), it is the Examiner, and not Appellants, who is "making an unsupported jump to an erroneous conclusion." The final sentence of section 9 reads: "Once a sufficient quantity of high quality seed key material (a few hundred bits) is available, strong computational techniques are available to produce cryptographically strong

sequences of unpredictable quantities for this seed material." The statement is unambiguous and clear. Eastlake is concerned with generating a few hundred random bits, and then uses those random bits, once, as a seed value for generating a ***sequence*** of random numbers. Eastlake does not teach, mention, or suggest use of a compressed data stream by a random number generator to continuously generate random numbers, extracting data from the compressed data stream to generate each next random number. Eastlake uses reversible compression to de-skew a skewed bit stream to produce a string or random bits. There is no other suggestion, in Eastlake, for using compression. Section 5.2.4 of Eastlake is the only section of Eastlake that discusses compression, and the compression is explicitly stated as being used to generate a random bit *string*, and not a compressed data stream.

With regard to section (vii), the Examiner's position appears to be that, although Saints does not teach, mention, or suggest use of compressed data or a compressed data stream, since Eastlake mentions compression, it would be obvious to combine compression with energy sampling to produce the currently claimed invention. This type of reasoning is exactly the type of conclusory and irrational reasoning that is stated, in *KSR*, as being insufficient to support an obviousness-type rejection. First, Eastlake does not teach, mention, or suggest use of compressed data streams, as discussed above, but only using compression to de-skew data collected from a data stream to produce a sequence of de-skewed data from which a relatively small number of random bits are extracted. Moreover, what purpose would compression possibly serve in Saints' system? Saints does not generate random-number-generator seeds. Saints generates the random data that Saints uses to generate random numbers by two mixing and hashing functions. There is no reason suggested by the Examiner for importing compressed data into Eastlake's random-number-generator unit. Data compression is computationally expensive. Designers do not add expensive and unnecessary data compression steps to devices. Neither Eastlake nor Saints teaches, mentions, or suggests a continuous random-number generation process in which data is extracted from a compressed data stream to generate each next random number. Eastlake uses a few random bits as a seed that is input to a random number generator and Saints does not teach, mention, or suggest data compression of any kind.

With regard to section (viii), Appellants again note that the filling of the pool is an initialization step. All that Saints discloses is that, on power-up, the pool is filled before random number generation begins. Claim 1, by contrast, includes the elements: "a monitor that receives the compressed data stream and monitors the compressed data stream to

determine whether or not sufficient data has been received in the compressed data stream to generate a next random number" and "a blocking switch controlled by the monitor to block output of a next random number by the random number generator when sufficient data to generate the next random number has not been received in the compressed data stream to generate a next random number." Saints makes no claim or assertion that the pool must be filled in order to generate a random number. Saints merely fills the pool, during initialization, before proceeding to generate random numbers on demand. Saints does not teach or suggest that for which the Examiner cites Saints. Saints does not teach, mention, or suggest use of a stream of compressed data.

With regard to section (ix), the relevant elements of claim 1 are: "a monitor that receives the compressed data stream and monitors the compressed data stream to determine whether or not sufficient data has been received in the compressed data stream to generate a next random number" and "a blocking switch controlled by the monitor to block output of a next random number by the random number generator when sufficient data to generate the next random number has not been received in the compressed data stream to generate a next random number." The language of claim 1 is clear. The monitor "monitors the compressed data stream to determine whether or not sufficient data has been received in the compressed data stream to generate a next random number," and the monitor controls a blocking switch "to block output of a next random number by the random number generator when sufficient data to generate the next random number has not been received in the compressed data stream to generate a next random number." Saints is entirely unrelated to these claim elements. Saints initializes a pool with energy samples that have been arithmetically combined with additional energy samples using a hash function. Then, Saints generates random numbers on demand. Saints does not teach, mention, or suggest anything related to monitoring a compressed data stream to determine whether or not sufficient compressed data is received to generate a next random number. There is no indication, in Saints, that the entire pool is needed to generate a random number. To the contrary, it appears that the contents of the pool are used as a circular buffer, and contain sufficient data to generate a large number of random numbers.

With regard to section (x), Appellants have explained, above, that the cited portion of Saints describes an initialization procedure for initializing the random number generation unit, and have nothing whatsoever to do with monitoring input of compressed data to block random number generation when insufficient compressed data has been received to

generate a next random number. Saints simply states that the pool is fully initialized prior to random number generation. This is clearly and explicitly stated by Saints. The cited passages do not state or suggest any kind of monitoring to ensure that sufficient compressed data has been received in order to generate a next random number. An analogy would be a text-processing program that searches a document for the first occurrence of the term "random." The program may read the entire document into memory, prior to undertaking the search, or may search for the term "random" as the document is being read into memory. The first approach does not, in any way imply, that the text-processing program includes a monitor that monitors input of data in order to determine whether sufficient text has been read into memory in order to search for the term "random." These are simply two different ways that a text-processing program may be initialized in order to begin searching. Saints simply fills the pool prior to commencing random number generation. Saints makes no indication or suggestion that a filled pool is required to generate a random number, and certainly includes nothing to indicate an on-going monitoring process. Saints simply carries out a power-on initialization that includes first filling the pool.

With regard to the phrase "initialization code," Appellants are simply drawing an analogy to initialization of processes that control automated systems. Indeed, Saints does not mention "initialization code," but, as discussed above, most of column 9 in Saints is devoted to describing power-up initialization of the random number generation unit. Furthermore, Saints also does not teach, mention, or suggest that for which Saints is cited by the Examiner. Filling a pool prior to launching random number generation in no way implies a monitor that monitors reception of compressed data to ensure that sufficient data is received to allow generation of a next random number and does not involve a blocking switch controlled by a monitor to prevent generation of a next random number when insufficient compressed data has been received.

With regard to section (xi), the Examiner's statement with regard to functionality is unsupported by citation to rule, case law, or statute. The current application explicitly shows the blocking switch, labeled "17" in Figure 1 and, in the paragraph that begins on line 9 of page 6, the blocking switch is discussed as utilizing a control signal output by the merge circuit 13. Saints does not teach, mention, or suggest a blocking switch of any kind. The Examiner is inferring that a blocking switch is present, because the Examiner feels that, since Saints does not begin generation of random numbers until power-on initialization is complete, there must be a blocking switch present that is "controlled by the monitor to

block output of a next random number by the random number generator when sufficient data to generate the next random number has not been received in the compressed data stream to generate a next random number." As discussed above, ***Saints does not in any way mention a blocking switch either as a circuit or as functionality.*** Saints' pool is a circular buffer that contains ample arithmetically mixed energy samples to generate numerous random numbers. Saints does not monitor filling of the pool to determine when sufficient energy samples are present to generate a next random number. Saints simply fills the pool, as part of power-on initialization, before launching on-demand random-number generations.

With regard to section (xii), Appellants again provide the two elements of claim 1 that define the monitor and blocking switch: "a monitor that receives the compressed data stream and monitors the compressed data stream to determine whether or not sufficient data has been received in the compressed data stream to generate a next random number" and "a blocking switch controlled by the monitor to block output of a next random number by the random number generator when sufficient data to generate the next random number has not been received in the compressed data stream to generate a next random number." These elements together describe a monitor that continuously monitors a stream of compressed data to ensure that sufficient compressed data has been received to generate a next random number. Appellants cannot understand why this relatively straightforward concept has presented such difficulty in interpretation. Blocking output of a next random number quite clearly disables random number generation, from the standpoint of a consumer of random numbers.

With regard to section (xiv), the Examiner's statements make no sense. The quality of a random number and cryptographically strong values are not currently claimed, and do not have any definite meaning, regardless of how often such phrases are used by Eastlake. Improving a random sequence generator does not necessarily imply anything about cryptographic strength or "quality" of random numbers. These statements are pointless, and have nothing whatsoever to do with the currently claimed invention.

With regard to section (xv), and as explained above, a stream of compressed data is a flow of compressed data, or a time sequence of compressed data. Time is inherent in a stream. Appellants are only trying to paraphrase the current claims in a way that will allow the Examiner to understand what is being claimed. The monitor clearly monitors the compressed data stream by monitoring data over time. Time is inherent in monitoring and in the compressed data stream.

With regard to section (xvi), the Examiner apparently fails to appreciate that simply detecting signals from sources does not constitute environmental sensing. Eastlake does not use the phrase "environmental," and does not provide a single sample of an environmental sensor. All of the sources of signals mentioned by the Eastlake are internal components of a computer that measure operational states of the internal components. Eastlake uses noise inherent in those signals. There is noise in many types of signals. Measuring noise due to resistance heating of an internal component of a system is not environmental monitoring or sensing. The current application makes it clear, as discussed above, that the environmental sensor measures fluctuation in an environmental parameter. In the first paragraph of page 4 in the current application, the current application discusses attempts by third parties to control environmental parameters in order to reduce uncertainty in sensor measurements and thus defeat encryption or other randon-number-depedent processes. Third parties might be able to alter the environment within which a device is located, but third parties cannot alter the sources discussed in Eastlake without disassembling the computer system or altering the computer system's operation. Environmental sensors have advantages that the internal sources of data discussed by Eastlake lack. For one thing, the environmental parameters measured by environmental sensors are always measurable. The rate of mouse clicks, internal heat generated by resistance, and disk-drive rotation are not as reliably available. By stating that "Eastlake discloses detecting input from a multitude of environmental sources (i.e. humans, software, hardware, physical phenomena)," the Examiner appears to be stating that anything and everything within a computer system is an environmental source, including stored data. The Examiner provides no support for this interpretation. Appellants respectfully suggest that the interpretation constitutes an attempt by the Examiner to disregard the term "environmental." However, the Examiner is not free to disregard claim language.

CONCLUSION

Embodiments of the currently claimed invention involve using a data stream, continuously produced by one or more environmental sensors, to continuously generate random numbers. However, embodiments of the present invention include a monitor that ensures that sufficient data has been received before allowing a next random number to be generated. Embodiments of the present invention compress the environmental-sensor data

into compressed data streams, and continuously extract data from the compressed data streams to generate random numbers. Neither Eastlake nor Saints teaches, mentions, or suggests a continuous process by which random numbers are generated from data extracted from a compressed data stream. Neither Eastlake nor Saints teaches, mentions, or suggests the use of environmental sensors. Neither Eastlake nor Saints teaches, mentions, or suggests the currently-claimed blocking switch that is controlled by the currently claimed monitor to disable random number generation until sufficient compressed data is received from the compressed-data stream to generate a next random number. Neither Eastlake nor Saints, alone or in combination, can possible make obvious the currently claimed invention, since the combination of Eastlake and Saints fails to teach, mention, or suggest a single element of independent claim 1.

Appellants respectfully submit that all statutory requirements are met and that the present application is allowable over all the references of record. Therefore, Appellants respectfully request that the present application be passed to issue.

Respectfully submitted,
Gadiel Seroussi et al.
OLYMPIC PATENT WORKS PLLC

By ______________________
Robert W. Bergstrom
Registration No. 39,906

Olympic Patent Works PLLC
P.O. Box 4277
Seattle, WA 98104
206.621.1933 telephone
206.621.5302 fax